

SYSTEM MODELING AND IMPLEMENTATION OF A GENERIC VIDEO CODEC

Jong-il Kim and Brian L. Evans*

Department of Electrical and Computer Engineering,
The University of Texas at Austin
Austin, TX 78712-1084
{jikirim,bevans}@ece.utexas.edu

Abstract -

The rapidly emerging and increasing complexity of video coding standards require a new design paradigm. This paper describes a modular, scalable, extensible simulation and design methodology for system-level design of video codecs.

Video codec dataflow is modeled by synchronous dataflow, and implemented in a heterogeneous CAD framework. As a result, generic video codec is decomposed into basic modules. Each module is easy to interface and extend. Any video codec standard (e.g., H.263+ or MPEG-4) can be mapped on that basis and be retargeted for various architectures including DSPs and ASIPs.

We develop module libraries for video codecs which can be dynamically linked to an extensible framework for simulation and algorithm development. Some parts of the basic modules can be mixed with other domain modules which may have different interaction semantics especially for hardware design and retargeting purposes. We based our framework on the Ptolemy software environment.

BACKGROUND

Multimedia industries are witnessing a rapid evolution toward integrating complete systems on a single chip. Standards for multimedia video codecs and teleconferencing are the significantly increasing in complexity and flexibility, and new ones are emerging on an annual basis. At the time that MPEG-1 chips were available on the market, the MPEG-2 standard was being finalized. Before the commercial market for the H.263 standard [1] begins, a new videoconferencing standard (H.263+) [2] emerges with multiple options and scalabilities. The continued evolution of video coding standards [3] from MPEG-1 to MPEG-4 is accompanied by manifold complexity increase with dynamic composition of audio-visual objects and algorithm tool boxes.

The conventional approach for system-level design and integration of video processing technology, combined with the level complexity and extremely short product development cycles, cannot find robust, cost-effective solutions

*This work was supported by the Accelerix Inc. and by the National Science Foundation CAREER Award under Grant MIP-9702707.

simply by implementing large parts of the system functionality in software running on application-specific instruction set processor (ASIP) cores.

To support simulation and its extension to synthesis for video encoders and decoders, this paper focuses on the system-level modeling and high-level abstraction of a generic video codec, and its implementation in a retargetable framework featuring mixed-domain simulation. Based on the model, new video compression and decompression standards can be implemented in a formal, consistent and extensible framework with well-defined and optimized processing primitives. The proposed design scheme is implemented and validated in Ptolemy [4].

VIDEO CODING STANDARDS

The motion picture experts group (MPEG) and international telecommunication union (ITU) have recommended several multimedia video coding standards since the late 1980s. From a technical point of view, video compression standards can be categorized into two groups. One is for low-complexity, frame-based coding schemes such as MPEG-2 and H.263. The other is for high-complexity, object-based coding algorithms such as MPEG-4, H.263+ [1, 3, 5]. In MPEG-2 and H.263, the video information is assumed to be rectangular, or of fixed size, displayed at a fixed interval. All of these standards have been applied to multimedia storage and communication such as CD-ROM, digital versatile disk (DVD), digital video broadcast (DVB), teleconferencing and high definition TV (HDTV).

The concept of a video object (VO) and video object plane (VOP) have been introduced and realized in MPEG-4. VO and VOP correspond to entities in the multimedia bitstreams that a user can access and manipulate (e.g., with cut and paste operations). The VOP can have arbitrary shape. At the decoder side, composition information is sent to indicate where and when each VOP is to be displayed along with the VOP itself. Also, the user at the decoder side may be allowed to change the composition of a displayed scene. H.263 and H.263+ can be applied for object-oriented coding with additional binary mask information, but the functionality is not so broad as that of MPEG-4.

An MPEG-4 core coder has a similar structure to H.263 or MPEG-2, and an MPEG-4 generic coder requires shape information to support arbitrary shape VOP. The overall MPEG-4 encoder blocks are composed of discrete cosine transform (DCT) and quantization which compresses the input VOP data utilizing spatial correlation of the texture, and the inverse operations (IDCT and inverse quantization) reconstruct the input VOP and store on frame buffer for motion estimation. The input VOP is compared with the best motion estimated block, and difference between the input and predicted block is coded if it generates a more compact code.

H.263+ is an extension of H.263 [1, 2], providing twelve new negotiable modes. These modes improve compression performance, support scalable bitstreams, and provide error resilience for mobile services. Some of the basic

coding techniques are common to MPEG standards, and the base layer of H.263 is bitstream compatible with MPEG-4 decoders. The functionality of an MPEG-4 coder includes other coding standards, and it will provide generic technology for multimedia communication services and applications.

NEW DESIGN PARADIGM

System modeling

Observing the fast emerging multimedia standards and the rapid growth of the complexity of the design, we suggest a new design paradigm for system-level designers. While system designers seek to shorten design cycles, improve quality, and reduce cost of the systems, no single design method is applicable to the entire system such as multimedia in that they contain a variety of algorithms implemented by many software and hardware technologies. Successful system design approaches integrate application-specific design methods and implementation technologies in a formal, consistent, extensible framework. The entire system can be specified, simulated, and synthesized in a formal, consistent framework, and extensibility supports new algorithms, tools, languages, and architectures to be integrated into the given framework to be rapidly retargeted to new technologies.

To support this new simulation and design paradigm, generic dataflow of video codec system is modeled by synchronous dataflow (SDF) and implemented in the heterogeneous framework such as Ptolemy. Using models of computation is key in supporting heterogeneity, which is the semantics of the interaction between modules or components in each domain [6, 7]. The design approach is based on the use of one or more formal models to describe the behavior of video codec system at a high level of abstraction. During the implementation process, many different specification and modeling techniques can be used [7]. Homogeneous synchronous dataflow (HSDF) model is composable because the number of token produced and consumed on each arc is one, so that HSDF model is adequate in simulation level which requires dynamic composing of the SDF libraries and modification of the algorithm. The hierarchical construction of the domain module allows us to implement the codec in more generic, flexible and efficient manner.

Module Structure

Figure 1 shows two modules, called stars in Ptolemy. At each firing, ReadVideoYUV and DisplayVideoYUV modules in Figure 1 read one frame YUV 4:1:1 format color image and display it, and ImageMagic visualization tool is invoked and consumes the image as soon as DisplayVideo block is fired. Among video codec modules, these two blocks correspond to source and sink blocks, which have only input or output port.

Since the firing rule obeys SDF semantics, unique behavior of a module is

described with minimal global variables. Since the behavior is compile-time predictable, each module can be optimized locally with inline code. Once a module is designed, as in Figure 1, it serves as a graphical programming tool as in a CAD platform. Therefore, module interconnections are simpler for a block consisting of a small number I/O ports.

Figure 2 illustrates the video codec modules. Each IO port is simplified but generalized so that it conveys general information through the interconnection. When we develop a module, there is a tradeoff between the composability and efficiency. Homogeneous SDF model is desirable to seek composable module design. A memory efficient module requires different scheduling algorithm and design metric.

Starting from the developed SDF libraries for generic video codec, hierarchical and mixed-domain simulation can be possible. Once the SDF model is fixed, the construction of efficient loop structure from SDF graphs allows the advantages of inline code generation under stringent memory constraints. A variety of efficient design metrics exist, so that during the modeling and optimizing the entire systems, different rules can be applied.

Integration and mixed simulation

From the composable SDF modules, given unique behavior for each, we can build a system in which each block comes back to its original state after a fixed number of firings. Figure 3 is a motion estimation and compensation system, where input, motion estimated, and motion compensated images are displayed one-by-one during each firing. Maintaining dynamic linkage with the platform, we can simulate and optimize a domain-specific algorithm. In a hierarchical or peer-to-peer way, we can mix the system, each having different semantics, which brings a wealth of design and simulation for various circumstances. Also global optimization is possible for a specific application through static scheduling algorithms in a given platform.

For the purpose of overall video codec system design, detail modeling for the video codec such as MPEG-4 and H.263 cannot be fixed but variable. If the video codec combines with some network or communication protocols, or if it involves multiple object scalability, discrete event model is better for exact simulation because of its additional time tag semantics. Ptolemy gives rich libraries for SDF and allows mixed model simulation and supports code generation in C and VHDL.

CONCLUSION

For rapidly growing multimedia video standards, we suggest a new system-level design paradigm, and validated in a heterogeneous CAD framework. We developed composable basic modules which are easy to interface with foreign tools, and flexibly controlled by other design criteria.

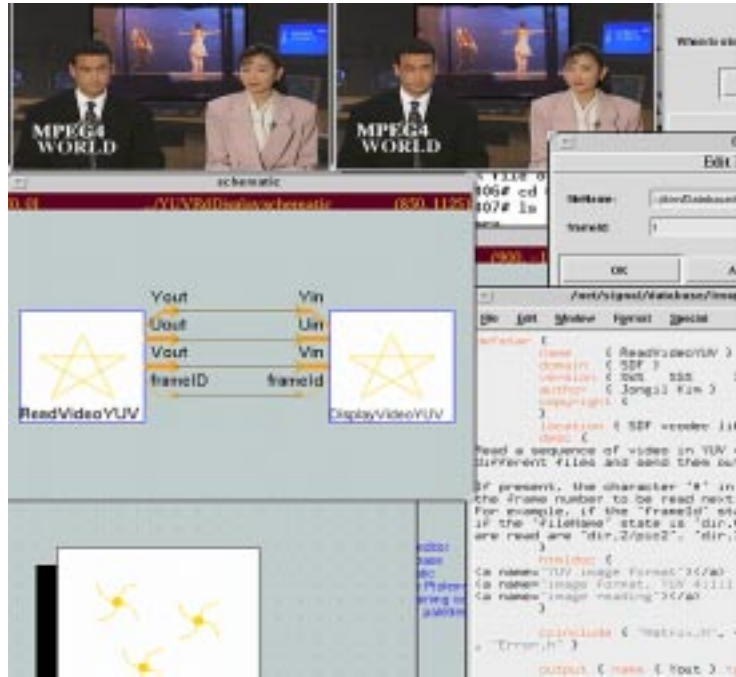


Fig. 1. An example of modular software development

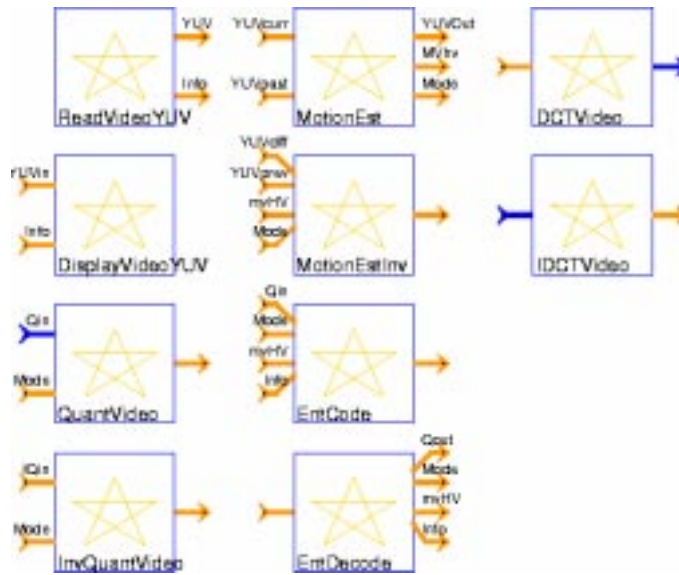


Fig. 2. Video codec blocks in SDF (synchronous dataflow) domain

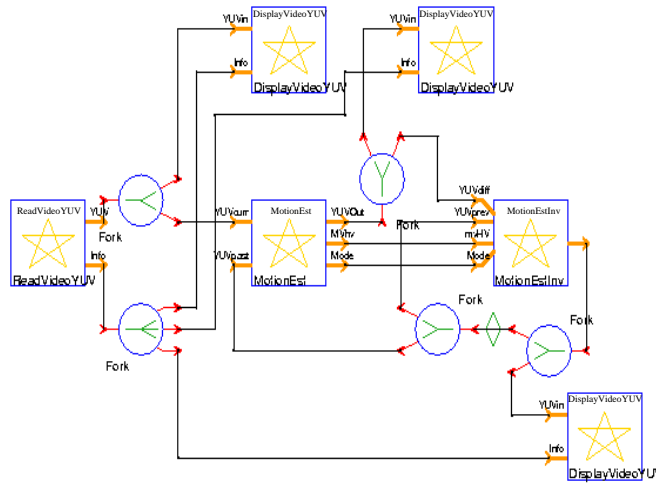


Fig. 3. An example of motion estimation and compensation system using video codec blocks

The developed video codec libraries are dynamically linked to an extensible framework for simulation, and once the modules are fixed, those libraries are statically linked. In Ptolemy platform, those developed modules are further exposed to various optimization metrics such as memory for embedded processor applications. The models of computation and the proposed design paradigm of video codec algorithm also helps automate partitioning of the design into hardware and software.

References

- [1] J. Hartung, A. Jacquin, J. Pawlyk, J. Rosenberg, H. Okada, and P. E. Crouch, "Object-oriented H.263 compatible video coding platform for conferencing applications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 1, pp. 42–55, Jan. 1998.
- [2] B. Girod, E. Steinbach, and N. Farber, "Performance of the h.263 video compression standard," *Journal of VLSI signal processing*, vol. 17, no. 2, pp. 101–111, Nov. 1997.
- [3] T. Sikora, "MPEG digital video coding standards," *IEEE Signal Processing Magazine*, vol. 14, no. 5, pp. 82–100, Sept. 1997.
- [4] S. S. Bhattacharyya, P. K. Murthy, and E. A. Lee, "Synthesis of embedded software from synchronous dataflow specifications," *to appear in Journal of VLSI Signal Processing*, <http://ptolemy.eecs.berkeley.edu/papers/publications.html/>.
- [5] L. Chiariglione, "MPEG and multimedia communications," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 7, no. 1, pp. 5–18, Feb. 1997.
- [6] P. K. Murthy, S. S. Bhattacharyya, and E. A. Lee, "Combined code and data minimization for synchronous dataflow programs," *Journal of Formal Methods in System Design*, vol. 11, no. 1, pp. 41–70, July 1997.
- [7] S. Edwards, L. Lavagno, E. A. Lee, and A. Sangiovanni-Vincentelli, "Design of embedded systems: Formal models, validation, and synthesis," *Proceedings of the IEEE*, vol. 85, no. 3, pp. 366–390, Mar. 1997.